International Journal of Advances in Data and Information Systems
Vol. 1, No. 1, April 2020, pp. 17-25
ISSN: 2721-3056, DOI: 10.25008/ijadis.v1i1.14

Analysis of Malware Impact on Network Traffic using Behavior-based Detection Technique

Adib Fakhri Muhtadi[1], Ahmad Almaarif[2]
[1,2]Department of Information System, Telkom University, Indonesia

Article Info

Article history:

Received Jan 18, 2020
Revised Feb 12, 2020
Accepted Mar 28, 2020

Keywords:

malware
dynamic analysis
behavior-based analysis
network traffic
API Call network

ABSTRACT

Malware is a software or computer program that is used to carry out malicious activity. Malware is made with the aim of harming user's device because it can change user's data, use up bandwidth and other resources without user's permission. Some research has been done before to identify the type of malware and its effects. But previous research only focused on grouping the types of malware that attack via network traffic. This research analyzes the impact of malware on network traffic using behavior-based detection techniques. This technique analyzes malware by running malware samples into an environment and monitoring the activities caused by malware samples. To obtain accurate results, the analysis is carried out by retrieving API call network information and network traffic activities. From the analysis of the malware API call network, information is generated about the order of the API call network used by malware. Using the network traffic, obtained malware activities by analyzing the behavior of network traffic malware, payload, and throughput of infected traffic. Furthermore, the results of the API call network sequence used by malware and the results of network traffic analysis, are analyzed so that the impact of malware on network traffic can be determined.

Corresponding Author:

Ahmad Almaarif,
Department of Information System,
Telkom University,
Jalan Telekomunikasi, Bandung, Indonesia.
Email: ahmadalmaarif@telkomuniversity.ac.id

1. INTRODUCTION

In the era of IoT (Internet of Things), one of the biggest threats on the internet today is malicious software, usually called malware because almost all of the main causes of internet security problems are malware. Malware is a program that was created with the intention of damaging it by breaking into a computer system. Malware has various types, namely viruses, worms, spyware, adware, trojans, keyloggers, rootkits, botnets and phishing [1]. Malware can be spread using various attack vector, including website and USB Flash Drive [2]. One example of malware is botnet, botnet is usually used to send spam and phishing host websites that make it difficult to track and blacklist [3]. In addition to botnets, malware that is often used to infiltrate a system is spyware. According to statistical data, 70-80% of spyware comes from websites that are considered safe by internet users [4].

Based on reports from ShadowServer, there are thousands of new malware samples that are received every day [3]. Each malware sample is analyzed one by one with the aim to find out what type of malware, how big the threat is and how to deal with it. Complete information of malware sample can be retrieved by using static analysis, a malware analysis conducted by examining the source code of malware [5]. However, for some cases this analysis is ineffective and inefficient because malware creators started using executable packing and obfuscation techniques, this technique can create several new malware variants from one malware [6]. In order to overcome this problem, dynamic analysis is needed.

There are a lot of threats against network traffic, including eavesdropping, IP Poisoning, ARP Cache Poisoning [7], and malicious network traffic containing malware and virus. Dynamic Analysis usually analyzes malware behavior such as network activity, API calls, file operations and registry modification records by executing samples in a virtual environment. The disadvantage of dynamic analysis is that this method requires a lot of time and resources for executing malware [8].

On Windows Operating System, every executable program needs to make a set of API calls. For example, for file management there are several API calls namely, OpenFile, DeleteFile, FindClose, FindFirstFile, GetFileSize [9]. Hence, API call information obtained after conducting dynamic analysis with behavior-based techniques is useful to find out what activities are carried out and what behaviors are owned by the malware so that it can be known what type of malware is and how much the threat is to the system based on the API call being run.

Analysis of malware on network traffic using dynamic analysis is necessary because there is not many previous researches has been conducted focusing on API Call [21]. To get the results of this analysis in a complex way, we need API call network for each malicious activity run by malware using the Cuckoo Sandbox tool. Network APIs obtained after executing malware in a virtual environment will be analyzed in order of their calls and what network APIs are used by the malware so that it can know the activities carried out by malware that utilize the Windows API especially the API network. After all the information about the network API is obtained, an analysis of network traffic is recorded while running malware in a virtual environment using Wireshark. Analysis carried out on the results of capturing network traffic is to look at the behavior of malware on network traffic, the payload carried by malware, and the measurement of traffic between normal traffic and malicious traffic that has been infected with malware [22].

Based on the results of the analysis of the sequence of network APIs and network traffic, the final result of this research is an explanation of how the impact of each malware sample used in this study on network traffic.

2. LITERATURE STUDY

2.1 Malware

Malware or Malicious Software is software that is made to attack by damaging, disturbing, retrieving important information that involves the confidentiality, integrity and availability of data of a system or application [10]. Malware is software that is explicitly designed to perform malicious activities such as Trojans, Viruses, Spyware and Exploit [11]. The way malware works is by entering into a system such as an existing computer system through various applications that exist on the system or it can also be through sending data from a device that has been exposed to a virus [12].

Malware can be divided into several types according to the method of operation and its characteristics [13], along with the types of malware.

1. Logic Bomb: malware that has two core parts namely, contents and triggers. This type of malware will be active if the trigger is run. Similar to time bombs, Logic Bombs can be active for a certain period of time.
2. Trojan Horse: a program that looks harmless but secretly runs malicious tasks. For example, when the user wants to do the login to a website by entering a username and password, but the website is already installed program password-grabbing, the role of the Trojan Horse is displaying a message error that states that the user entered your username and password, but behind the processes that the Trojan Horse is already taking information user authentication.
3. Back Door: a mechanism that goes through security standard inspection processes such as authentication.
4. Virus: a malware that will multiply itself by infecting other programs that are running.

5. Worm: malware that is similar to a virus that can copy itself, the difference is that a worm can copy itself from one system to another over a network and does not depend on executable programs.
6. Rabbit: a program that uses all resources on a system. An example is the Fork Bomb which always makes a new process repeatedly with an infinite number of loops so that makes the system slow.
7. Spyware: a program that takes a computer's information and sends it to other people.
8. Adware: a program similar to Spyware but the focus is marketing advertisement.
9. Zombies: a term for a computer that has been attacked / hacked without the user's knowledge. Usually the activity carried out is to spread spam email.

2.2 Dynamic Analysis

Malware Analysis is a collection of processes carried out by dissecting malware to understand how it works, how to identify it and how to defeat or eliminate it. This process is needed to adapt the development of malware detection techniques to the growth of new malware. Malware Analysis is divided into three methods namely Static Analysis, Dynamic Analysis and Hybrid Analysis [14].

Dynamic Analysis is a method of detecting malware by running the malware in a virtual environment. This method monitors the behavior of malware, its interactions with the system and its effects on the system. Analysis with this method is required to provide an isolated environment so as not to cause other programs to be affected by running malware directly [15].

2.3 Behavior-based technique

Behavior-based is a technique of the Dynamic Analysis method whose purpose is to determine the behavior of malware. This technique is carried out by executing malware and monitoring its behavior in network activity, API calls, file operations and registry modification notes. The drawback of this technique is that it takes resources and time for many, while the advantages of this technique is able to identify or introduction of new malware [16].

Behavior-based technique has several advantages and disadvantages [17]. The following are advantages and disadvantages of behavior-based technique. The advantages are:

- Detecting malware behavior is complex because this technique detects malware from behavior to the final destination of malware.
- Can identify new malware.
- Can identify a lot of malware with less time compared to other techniques.
- Simplify the classification of malware because it uses several tools to automate detection.

On another hand, the disadvantages of behavior-based technique are:

- Does not detect in detail such as checking for signatures and bits of malware.
- Couldn't find a malware solution.
- It consumes a lot of resources because this detection has to run malware directly in an environment.

2.4 Windows Network API

APIs or Application Programming Interfaces are application programming interfaces created by Windows in order to interact with the kernel system. For example, if the application wants to run File Management, the activity will involve several API calls such as OpenFile, DeleteFile, FindClose, FindFirstFile, GetFileSize. There are two API calls that will be used in this research, they are memory API and network API. The Windows API is dynamic-link libraries (DLLs) that are part of the Windows operating system. The advantage of using the Windows API is that it can save development time because it contains dozens of functions already written and waiting to be used. Network API is an API in the Windows operating system that is used in the process of transfer/request data in the network, while also allowing communication between applications through the network. Network

API is usually used by malware by hiding malicous code into sending traffic or requesting data on a network [18].

2.5 Network Analysis

Network analysis usually includes traffic analysis, protocol analysis, packet analysis, eavesdropping, etc. Network analysis is the process of capturing a network of traffic and checking it in detail to find out what is happening on the network. A network analyzer is a combination of hardware and software. In this study, the network analyzer used is Wireshark.

Wireshark is one of the best network analyzer applications available free. Wireshark has many advantages such as a good graphical user interface (GUI), supports 400 protocols, and is active in development and maintenance. Wireshark can be run on various operating systems such as UNIX-based systems, Mac OS X, and Windows [19].

3. RESEARCH METHOD

This research method is explained by the conceptual model. The conceptual model in this study consists of several important components, namely problems, environment, concepts, methods, IT artifacts, and evaluation. The problem and environment component is one part that forms the basis of research. The problem component contains the factors that are the reasons for this research. The environmental component is the party or part involved in this research such as people and technology. Technology used in this research are malware dynamic analysis tools, Windows 7 OS, virtualization software, Cuckoo Sandbox, and network traffic. Artifacts of this research is analysis of malware impact on network traffic while evaluation method used is malware dynamic analysis

4. RESULTS AND DISCUSSION

Testing is conducted through network API analysis and network traffic. The first test was performed on the Cuckoo Sandbox that was installed on VMware. Using Cuckoo, information about all network APIs used by malware is obtained. The next test is testing using Wireshark. Wireshark is a network traffic analysis tool intended to monitor network traffic. This test aims to obtain the activity carried out by malware on network traffic so that it can be seen how the behavior of malware, payload, and its impact on network traffic. In this test, the analysis carried out is divided into three parts, namely the analysis of network traffic to determine the behavior of malware, analysis of the payload to determine the activities carried out, and analysis of throughput measurements to determine its impact on network traffic.

4.1 Cuckoo Sandbox (API network)

After analyzing Cuckoo for all malware samples, network API sequences can be grouped. Grouping is done based on TID (task ID) for each network API. Next is the grouping of API network sequences after analyzing 30 malware samples.

Table 1. List of API *Network*

Kelompok	Urutan API *network*
API 1	*InternetCrackUrlA-InternetOpenA-InternetConnectA-HttpOpenRequestA-HttpSendRequestA-InternetReadFile-InternetCloseHandle*
API 2	*WSAStartup-socket-setsockopt-closesocket-GetAddrInfoW*
API 3	*getaddrinfo-socket-connect-send-recv-closesocket*
API 4	*WSAStartup-InternetOpenA-InternetConnectA-socket-closesocket-NSPStartup-socket-closesocket-WSASocketA-bind-setsockopt-ConnectEx-shutdown-closesocket-InternetCloseHandle-URLDownloadToFileW*
API 5	*WSAStartup-socket-setsockopt-closesocket-WSASend-WSARecv-shutdown-closesocket-HttpQueryInfoA*
API 6	*WSAStartup-WSASocketW-setsockopt-closesocket-gethostname-gethostbyname-GetAdaptersAddresses-GetAdaptersInfo*

API 7	*WSASocketW-WSAConnect-ioctlsocket-closesocket*
API 8	*WSAStarup-InternetCrackUrlA-ObtainUserAgentString-socket-gethostbyname-connect-closesocket*
API 9	*Socket-ioctlsocket-gethostbyname-connect-select-closesocket*
API 10	*InternetOpenA-InternetOpenUrlA-InternetReadFile-InternetCloseHandle*

Using API Call listed on Table 1, experiment is conducted resulted in API network used the most as depicted in Figure 1.

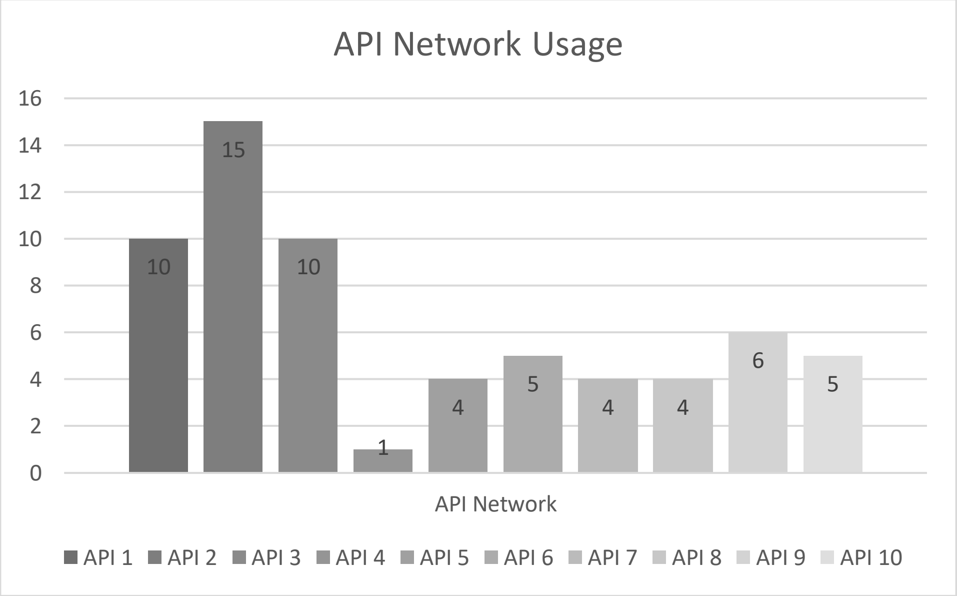


Figure 1. Graph of API network groups usage

Based on the chart above, we can conclude that the API group network the most widely used is API Network Group 2, *WSAStartup-socket-setsockopt-closesocket-GetAddrInfoW*. This means that half of the 30 malware samples have functions that can use the Windows Socket API to provide connection-oriented two-way communication between client and server malware. With the Windows Socket API, malware servers can affect Windows networks such as Quality of Service (QoS) so that an infected network can decrease QoS on network traffic.

In the second place, the most common use of the API network group is API 1 and 3 groups with a total of 10 malware samples each. It can be said that as many as 10 malwares with the API Network Group 1 have activities to access a URL or IP and then provide file transfer services between client and server so that, without realizing it, the server can send and retrieve data on the client computer. Whereas 10 malwares with API Network Group 3 have activities to send and receive data through a predetermined socket. The API Network Group 3 will affect the throughput of network traffic because the process of sending and receiving data happens.

4.2 Wireshark

4.2.1 Malware Behavior in Network Traffic

The testing aims to determine the behavior of malware on network traffic. Following is the malware behavior on network traffic which is grouped into four parts, namely DNS and NetBIOS requests, DNS and rDNS, ICMP echo, suspicious activities [20].

- B1: make a DNS request / rDNS / NetBIOS name request.
- B2: Connection attempt to IP address failed while DNS request was successful.
- B3: attempt to connect to an IP address in the absence of rDNS.

- B4: Connection attempt to IP address with rDNS failed.
- B5: ICMP echoes IP addresses without replies or error messages.
- B6: try a TCP connection to an IP address that has a successful message at ICMP echo request.
- B7: use TCP connections to IP addresses that have never been used in DNS, NetBIOS, ICMP.

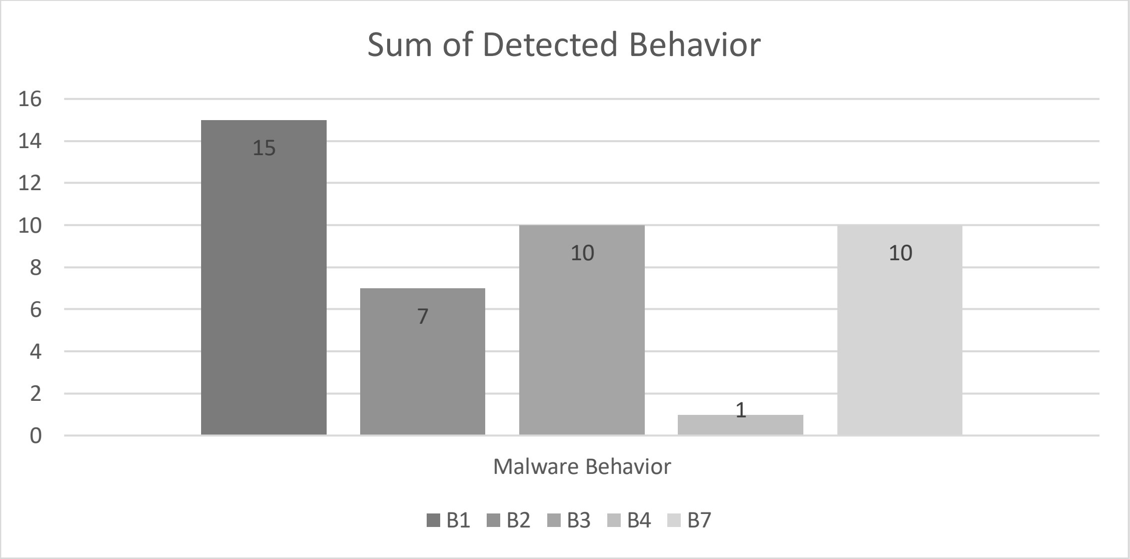
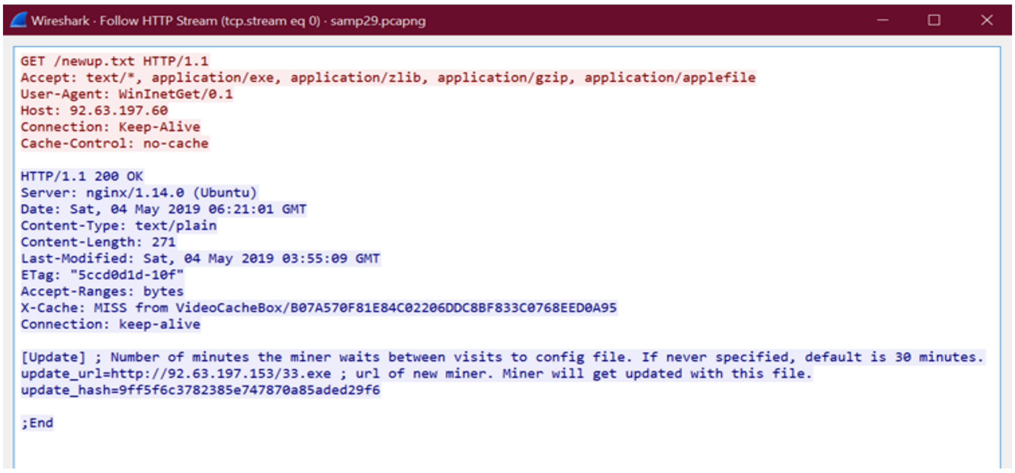


Figure 2. Malware Behavior

From Figure 2, the most detectable *malware* behavior is B1 or the first behavior. A total of 15 *malware* samples were detected on B1. This is because half of the 30 *malware* samples make DNS *requests* to obtain IPs from each *domain* that the *malware* wants to connect to.

4.2.2 Payload

This experiment aims to determine the activities carried out by *malware* on network traffic. Here is a test from one of the samples


Wireshark · Follow HTTP Stream (tcp.stream eq 0) · samp29.pcapng

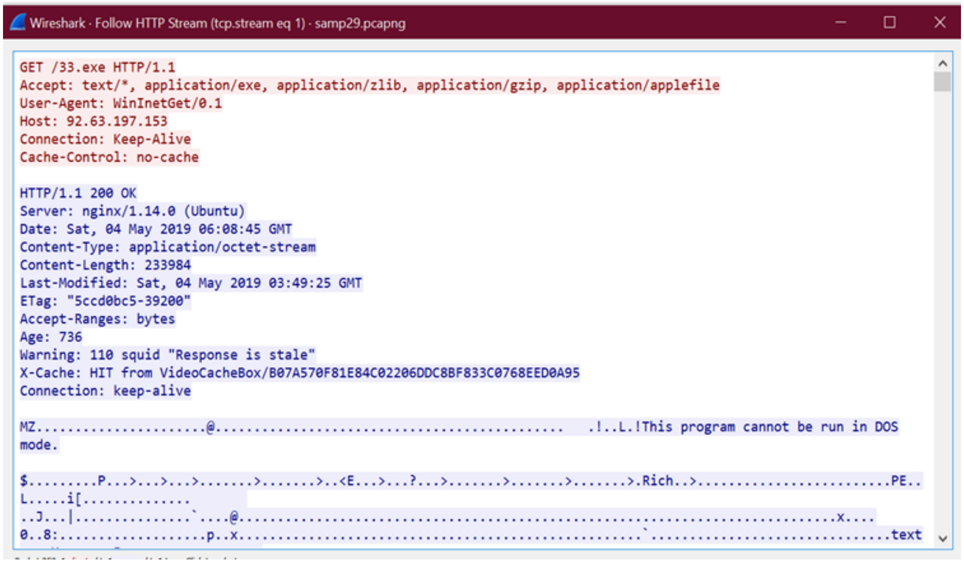
```
GET /newup.txt HTTP/1.1
Accept: text/*, application/exe, application/zlib, application/gzip, application/applefile
User-Agent: WinInetGet/0.1
Host: 92.63.197.60
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sat, 04 May 2019 06:21:01 GMT
Content-Type: text/plain
Content-Length: 271
Last-Modified: Sat, 04 May 2019 03:55:09 GMT
ETag: "5ccd0d1d-10f"
Accept-Ranges: bytes
X-Cache: MISS from VideoCacheBox/B07A570F81E84C02206DDC8BF833C0768EED0A95
Connection: keep-alive

[Update] ; Number of minutes the miner waits between visits to config file. If never specified, default is 30 minutes.
update_url=http://92.63.197.153/33.exe ; url of new miner. Miner will get updated with this file.
update_hash=9ff5f6c3782385e747870a85aded29f6

;End
```

Figure 3. Payload 1



Wireshark · Follow HTTP Stream (tcp.stream eq 1) · samp29.pcapng

```
GET /33.exe HTTP/1.1
Accept: text/*, application/exe, application/zlib, application/gzip, application/applefile
User-Agent: WinInetGet/0.1
Host: 92.63.197.153
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sat, 04 May 2019 06:08:45 GMT
Content-Type: application/octet-stream
Content-Length: 233984
Last-Modified: Sat, 04 May 2019 03:49:25 GMT
ETag: "5ccd0bc5-39200"
Accept-Ranges: bytes
Age: 736
Warning: 110 squid "Response is stale"
X-Cache: HIT from VideoCacheBox/B07A570F81E84C02206DDC8BF833C0768EED0A95
Connection: keep-alive

MZ......................@..........................................   .!..L.!This program cannot be run in DOS
mode.

$........P...>...>...>.......>.......>..<E...>...?...>.......>.......>.......>.Rich..>.......................PE..
L.....i[.............
..J...|..............`....@...........................................................................x....
0..8:..................p..x..........................................`.............................text
```

Figure 4. Payload 2

Figure 3 and Figure 4 are payload contains HTTP request made by samp29.exe. The sample accesses two files named newup.txt and 33.exe. From Figure 3, the first payload belongs to newup.txt. In the payload there is a command that directs the client to access the 33.exe file so that the HTTP request activity against 33.exe also occurs. Figure 4 is the payload owned by 33.exe. In the payload, there is an MZ code at the beginning of the payload. MZ is the code on Wireshark with a function to download a file with format executable. This can be proven by exporting the file via the File menu - Export Object - HTTP in Wireshark.

4.2.3 Throughput Analysis

In testing this section, aims to determine the impact of malware on network traffic, especially the speed of traffic after being infected with malware. To measure throughput in this analysis, a file test with an executable format is downloaded. This file is 15.17 MB in size.

To measure throughput, the data taken is the number of average bit/s. This data shows the amount of bandwidth used when downloading files that have been provided. In addition, the data shows the size of the data entering the traffic. The results of bandwidth testing can be seen on Table 2.

Table 2 Throughput Measurement Result

Sample	*Size (Bytes)*	*Throughput (Kbits/s)*
Normal Traffic	17339072	4379
Malware Infected Traffic	17360366 - 22023405	1138 - 3716

Table 2 describe the normal traffic throughput reaches 4379000 bits/s or 4.3 Mbps. While malware traffic has a throughput between 1138-3716 Kbits/s. In addition, data that was successfully recorded on normal traffic were 17339071 Bytes. While data on malware traffic has a larger size.

4.3 Impact on Network Traffic

Based on the results of malware analysis with the API network and network traffic, it can be seen how the impact of malware on network traffic. The impact contained in the table below is a significant impact from different types of malware samples.

Table 3 Impact of Malware on Network Traffic

Sampel	Impact
Samp30.exe	This sample aims to create or look for weaknesses in the system and then access websites that have been infected with malware. As a result the throughput has decreased.
Samp29.exe	These samples do evil activity by downloading a malware that type of Trojan Chapak. This type of malware has other Windows API functions that attack the system including network traffic, resulting in decreased throughput.
Samp28.exe	This sample downloads bits that are suspected to be malicious code. The decrease in throughput can be caused by the downloading process or malicious code that has invaded the system.
Samp27.exe	This sample creates an IP backdoor connected to the payload that is suspected to be the entrance to a malicious file. By utilizing the Windows Socket API, this sample can affect network traffic QoS, thereby causing a decrease in throughput.
Samp23.exe	This sample is categorized as a sample that conducts mining with socket minelitecoin.com:8336 as its connection path. This process takes a considerable amount of bandwidth but the size of data recorded during traffic recording cannot be detected.

5. CONCLUSION

Based on the results of the analysis of the impact of malware on network traffic using behavior-based detection techniques that have been done, the following conclusions are obtained:

a. Analysis of the impact of malware on network traffic using behavior-based detection techniques can be done by looking at the behavior of malware through its activity. The malware activity can be known by arranging the network API sequences used by malware and analyzing network traffic that has been infected with malware.

b. Malware activity can be identified by looking at the network API sequences used by malware. This sequence can be grouped by looking at the Task ID of each process that is running. In one process requires several network API functions so that the activity cannot be determined by only identifying each network API.

c. Malware activity can be determined through analysis of network traffic such as payload analysis, bandwidth usage, and malware network behavior. Payload analysis aims to see the activity of a malware by checking the functions or behaviors shown during testing. Throughput analysis aims to examine the impact of malware on network traffic. While the analysis of malware network behavior aims to detect whether the sample can be categorized as malware or not.

REFERENCES

[1] Saeed, I. A., Selamat, A., & Abuagoub, A. M. (2013). A survey on malware and malware detection systems. International Journal of Computer Applications, 67(16).

[2] Efendy, R. A., Almaarif, A., Budiono, A., Saputra, M., Puspitasari, W., & Sutoyo, E. (2019, November). Exploring the Possibility of USB based Fork Bomb Attack on Windows Environment. In 2019 International Conference on ICT for Smart Society (ICISS) (Vol. 7, pp. 1-4). IEEE.

[3] Bayer, U., Comparetti, P. M., Hlauschek, C., Kruegel, C., & Kirda, E. (2009, February). Scalable, behavior-based malware clustering. In NDSS (Vol. 9, pp. 8-11).

[4] Jain, M., & Bajaj, P. (2014). Techniques in detection and analyzing malware executables: a review. International Journal of Computer Science and Mobile Computing, 3(5), 930-935.

[5] Ismail, J.,"Static Method Malware Analysis | Jul Ismail, "2016. [Online]. Available: https://julismail.staff.telkomuniversity.ac.id/ analyzed-malware-metode-statik/. [Accessed: 21-May-2019]

[6] Perdisci, R., Lee, W., & Feamster, N. (2010, April). Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces. In NSDI (Vol. 10, p. 14).

[7] Almaarif, A., & Yazid, S. (2018). ARP Cache Poisoning sebagai Teknik Alternatif untuk Membatasi Penggunaan Bandwidth berbasis Waktu. Jurnal Rekayasa Sistem & Industri (JRSI), 5(02), 108-113.

[8] Kim, Y. S., Wang, E., & Rho, H. M. (2001). Geometry-based machining precedence reasoning for feature-based process planning. International Journal of Production Research, 39(10), 2077-2103.

[9] Merialdo, G., "Medusa," *Rev. Homeopath Medica.* , vol. 5, no. 2, pp. 61–62, 2012.

[10] Mell, P., Kent, K., & Nusbaum, J. (2005). Guide to malware incident prevention and handling (pp. 800-83). Gaithersburg, Maryland: US Department of Commerce, Technology Administration, National Institute of Standards and Technology.

[11] Cahyanto, T. A., Wahanggara, V., & Ramadana, D. (2017). Analisis dan Deteksi Malware Menggunakan Metode Analisis Dinamis. JUSTINDO, Jurnal Sistem & Teknologi Informasi Indonesia, 2(1), 19-30.

[12] Utama, W., "What Is Malware, Understanding, Explanation and Types of Malware That Need to Watch Out for," 2017. [Online]. Available: https://www.klikmania.net/apa-itu-malware. [Accessed: 21-May-2019]

[13] Aycock, J. (2006). Computer viruses and malware (Vol. 22). Springer Science & Business Media.

[14] Sikorski, M., & Honig, A. (2012). Practical malware analysis: the hands-on guide to dissecting malicious software. no starch press.

[15] Shijo, P. V., & Salim, A. (2015). Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804-811.

[16] Liu, W., Ren, P., Liu, K., & Duan, H. X. (2011, September). Behavior-based malware analysis and detection. In 2011 first international workshop on complexity and data mining (pp. 39-42). IEEE

[17] Jacob, G., Debar, H., & Filiol, E. (2008). Behavioral detection of malware: from a survey towards an established taxonomy. Journal in computer Virology, 4(3), 251-266.

[18] Webi, "Windows API Index - Windows applications | Microsoft Docs, "2018. [Online]. Available: https://docs.microsoft.com/en-us/windows/desktop/apiindex/windows-api-list#networking-and-internet. [Accessed: 21-May-2019]

[19] Orebaugh, A., Ramirez, G., & Beale, J. (2006). Wireshark & Ethereal network protocol analyzer toolkit. Elsevier.

[20] Morales, J. A., Al-Bataineh, A., Xu, S., & Sandhu, R. (2010, September). Analyzing and exploiting network behaviors of malware. In International conference on security and privacy in communication systems (pp. 20-34). Springer, Berlin, Heidelberg.

[21] Suryati, O. T., & Budiono, A. (2020). Impact Analysis of Malware Based on Call Network API with Heuristic Detection Method. International Journal of Advances in Data and Information Systems, 1(1), 1-8.

[22] R. N. Romli, M. F. Zolkipli, A. Al-Ma'arif, M. R. Ramli, and M. A. Salamat, "Understanding the Root of Attack in Android Malware", *International Journal of Integrated Engineering*, vol. 10, no. 6, Nov. 2018.